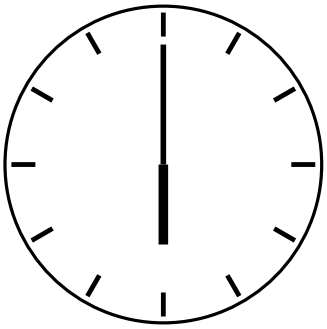
HKUST Programming Contest 2004 Fall

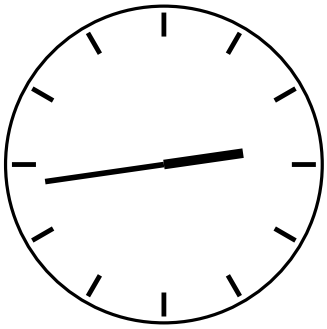
Date: October 16th, 2004
Time: 13:00 – 17:00
Venue: CS Lab 4

Problem A – Clock (again??)

There are two hands (pointers) on the watch. The hour-hand (usually thinker and shorter) indicates the hour of current time. The minute-hand (usually thinner and longer) indicates the minute of current time. In this problem, your will be given a specific time. You need to calculate after how many seconds, these two hands become opposite (i.e. the angle between these two hands is 180°). You can assume both hands of the watch move perfectly smooth.



06:00



02:43:38.181818…

Input

The first line of input contains a single integer, N, which specifies the number of test cases. For next N lines, each line contains the time in string format hh:mm:ss, where $01 \le hh \le 12$, $00 \le mm \le 59$ and $00 \le ss \le 59$.

Output

For each input, you need to calculate after how many seconds, these two hands become opposite. The output should be in exact value. If the answer is an integer, print out the integer. If the answer is greater than 0 and less than 1, print it out in the format "`p/q`" where p and q are two integers and relatively prime. If the answer is greater than 1 but not an integer, print it out in the format "`n+p/q`" where p and q are two integers and $p < q$ and they are relatively prime. i.e. if the answer is 4.5, the output should be "`4+1/2`", both "`9/2`" and "`4+5/10`" are not acceptable. Since the answer must be rational number, the description above covers all possible cases.

Sample Input

```
4
05:00:00
06:00:00
02:00:00
02:43:38
```

Sample Output

```
3600
0
2618+2/11
2/11
```

Problem B – Romeo and Juliet

Let's skip the story. In this problem, you are given a rectangular maze. Both Romeo and Juliet are placed inside the maze and you need to find out the least possible unit of time for them to meet inside the maze. The following are the detail definitions of the problem:

1. Romeo and Juliet are placed in different locations inside the rectangular maze initially.
2. Romeo and Juliet can meet if and only if they can go to the same location inside the maze at some time.
3. They cannot leave the rectangular maze.
4. There are four kind of land inside the maze: free space, lake, mountain, and wall.
5. Both Romeo and Juliet can walk through free space.
6. Both Romeo and Juliet cannot walk through wall.
7. Romeo doesn't know how to swim. i.e. he cannot pass through lake.
8. Juliet is not strong enough to pass through mountain.
9. It takes 1 unit of time to travel from one block to another block inside the maze.
10. The initial locations of them are free spaces.
11. They have the complete information about the maze and know the initial location of each other.

Input

The first line of input contains a single integer, N, which specifies the number of test cases. N cases are followed. The first line of each case contains 2 integers m, n where m is the number of rows of the maze and n is the number of columns of the maze and $1 \le m, n \le 20$. Each of next m lines contains a string of length n. These m strings specify the complete information of the rectangular maze as follow:

1. '#' is a block of wall.
2. '`.`' is a block of free space.
3. '`~`' is a block of lake.
4. '`^`' is a block of mountain.
5. '`R`' is the initial location of Romeo.
6. '`J`' is the initial location of Juliet.

Output

For each case, print out the least unit of time for them to meet. If that is impossible, print out the string "`Impossible.`".

Sample Input

```
7
1 2
RJ
1 6
R....J
1 7
R.....J
1 8
R.^..~.J
1 9
```

```
R.~...^.J
1 10
R~~~~~~~~J
10 16
.#..~~~.........
.#..####.#######
R#.....#.#...#.#
^#.....#...#....
^#^^^^.########.
.#.~~~.#..^^.^^.
.#.^^^^#..#####.
.#....J#..~.~.~.
.###############.
................
```

Sample Output

```
1
3
3
4
Impossible.
9
32
```

Problem C – Coin and Dice

Alice: “Hey, Bob, I’m too bored. Let’s play some games.”
Bob: “Let me flip a coin and you guess whether it’s head or tail.”
Alice: “Okay, better than idling.”
Bob: “How about I roll a dice and you guess the number?”
Alice: “Hmm… I don’t like ‘6’. ‘6’ means evil.”
Bob: “If I roll a ‘6’, I roll again until it’s not ‘6’. Is this alright?”
Alice: “Okay, let’s play both games now! I hope I can make all guesses correct.”
Bob: “…”
Alice: “What’s wrong?”
Bob: “Do you know that it’s impossible to make all guesses correct?”
Alice: “Why?”
Bob: “The probability is just too low.”
Alice: “What? What are you talking about?”
Bob: “Forget it. Anyway, let’s start.”

Then, two bored students start to play these two such stupid and boring games. And you won’t get an excited job either because your stupid task is to compute the probability that Alice cannot make all her guesses correct. Assume both the coin and the dice are fair.

Input
The input of each case is a line containing two non-negative integers, m and n, where $0 \leq m, n \leq 100$. m is the number of times Bob flipped the coin. n is the number of times Bob rolled the dice. The input is terminated by a line with two 0’s.

Output
For each case, print out the exact value of the probability that Alice cannot make all her guesses correct.

Sample Input
```
0 1
0 2
1 0
2 0
1 1
2 2
60 60
0 0
```

Sample Output
```
0.8
0.96
0.5
0.75
0.9
0.99
0.999999999999999999999999999999999999999999999999999999999999
```

Problem D – CyberCube

Martin and Roger awaken in a maze of mysterious rooms (Don't ask the author - why). Each room has six doors and looks identical. The doors are locked, but fortunately that they found some keys (labeled with numbers) near them. Soon they discover some more interesting things; it seems that the basic laws of physics don't apply in this world. That is the rules they know:

1. All the doors could be opened by any keys, but the room behind would be change according to the key.
2. Once they entry a room, they can't go back. The door will be locked automatically.

The world also exist some rules that they don't know:

1. Each room is labeled by a number (1 to 2147483648)
2. There have two kinds of keys: Gold and Silver.
   i. Gold key opens a door to connect to a room which its number is the multiple of current room number and key number.
   ii. Silver key opens a door to connect to a room which its number is the addition of current room number and key number.
3. If they go into a room which its number exceed 2147483648, they will be trapped and won't be able to escape from that room.

In the beginning, they are fallen into different room. Martin found 9 keys labeled from 1 to 9 and all are gold keys, while poor Roger could only find a key labeled with one and it is silver.

Martin and Roger are very hurry to find each other. (Don't ask why) Could you help to write a program to tell whether they can meet in this maze and how much time needed?

You may assume Martin is always fallen to room number one. And they would not stay in a room more than a unit of time. The time to transfer between two rooms can be neglected.

Input

The first line of input contains a single positive integer t, the number of test cases. Each test case contains a single integer n ($1 \leq n \leq 214783648$), which is the initial room number of Roger resisted.

Output

For each test case, output one line containing a single integer that holds the minimum unit of time for Martin and Roger to meet. If it is impossible, print a string "IMPOSSIBLE" instead.

Sample Input

```
5
12
162
100000
```

```
2000000000
2147483640
```

Sample Output

```
2
6
352
376000
IMPOSSIBLE
```

Problem E – XYZ Puzzle

XYZ Puzzle is a game board consists of a sequence of positive integers. You could select any number except the head and tail number in the board. The selected number is removed from the board. The score for each removal is equal to the sum of the number selected with its neighbor.

Example:

Initial configuration:
1 2 3 4 5
Available move: {2, 3, 4} Score = 0

{3} is removed.
1 2 4 5
Available move: {2, 4} Score = 2+3+4 = 9

{3, 4} is removed.
1 2 5
Available move: {2} Score = 9 + 2 + 4 +5 = 20

1 5
{2, 3, 4} is removed. No available move.
The game finished. Total score = 20 + 1 + 2 + 5 = 28

You are asked to write a program to find out the maximum score could be gain for different configuration of the board.

Input
The input file will contain one or more test cases, one per line.
Each test case contains n ($3 \le n \le 10$), the total number of integers in the board. It is followed by n integers, k_1, k_2 , ... k_n ($1 \le k_i \le 100$, for $1 \le i \le n$). If n = 0 it signals the end of the input.

Output
For each test case, print a line contains the maximum score.

Sample Input
```
5 1 2 3 4 5
5 2 1 5 3 4
6 30 20 40 50 70 60
0
```

Sample Output
```
30
31
570
```

Problem F – Death Note-book

Ratio Yager is a very clever high school student. One day he finds a mysterious notebook computer, which can be turned on without any internal battery or external source of power! However, only one program could be run on this computer, it is called - "DEATH".

DEATH User Guide:

The human whose name is entered shall die.

The name must be written by upper case alphabetic character which may contain up to 80 characters in length. (No extra white spaces at the beginning and the end.)

If the cause of death is not specified, the person will simply die of a heart attack.

If the time of death is not specified, the human die after 40 minutes of input. You may change the time of death by enter the name again.

If you write the time of death after the name like this
"<name> died at <time>" then that will happen. (All time will be in hh:mm 24hours format)

The cause of death could be specified by a command like this "<name>: killed by <name>". Victim will be killed by himself or other living people. You could specific the cause of death more than once, but only the last one before execution would take effect.

The death will not be realized unless it is physically for that human or it is reasonably assumed to be carried out by that human or murder. You couldn't kill a human twice, and ask a dead body to kill living people.

It could holds 100 lines of input.

You are asked to write a program to simulate the operation of DEATH, and then output the sequence of death

Input
There will be a number of lines of input; each line will contain a DEATH command specified in the user guide. Before each command, it will have a timestamp in 24 hours format to specific the input time. You could assume the input is sorted by the timestamp. The last line in the input will contain just the three character sequence "END", which should not be processed as a DEATH command.

Assumption:
1. No people in this world share the same name.

2. The time is started at 00:00
3. DEATH can not function after 23:59

Output
You should print out the entire death body name with their time of death. The output must be sorted by their time of death and then their name in alphabetical order.

Remarks: If more than one person dies in the same minute of time, the sequence of death is determined by their name in alphabetical order.

Sample Input
```
07:40 TOGASHI YOSHIHIRO
07:40 X died at 13:50
07:50 X: killed by KIRA
08:20 TOGASHI YOSHIHIRO: killed by X
11:30 KIRA
11:50 LOBSTER: killed by KIRA
12:00 LOBSTER died at 12:10
13:10 TOGASHI YOSHIHIRO
14:00 Y died at 14:00
14:00 Y: killed by Z
END
```

Sample Output
```
TOGASHI YOSHIHIRO died at 08:20
KIRA died at 12:10
Y died at 14:00
```